U S DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER ROZENBERG=1
TRANMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371	U.S. APPLICATION NO (If known,see 37 CFR 1 5) **09/889627**

INTERNATIONAL APPLICATION NO. PCT/IL00/00769	INTERNATIONAL FILING DATE 19 November 2000	PRIORITY CLAIMED 22 November 1999

TITLE OF INVENTION
METHOD FOR AUTOMATICALLY CONSTRUCTING A COMMAND INTERFACE

APPLICANT(S) FOR DO/EO/US
Dominique ROZENBERG et al.

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. [X] This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. [ ] This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. [X] This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. [ ] The US has been elected in a Demand by the expiration of 19 months from the priority date (PCT Article 31).
5. [X] A copy of the International Application as filed (35 U.S.C. 371(c)(2))
   a. [ ] is attached hereto (required only if not transmitted by the International Bureau).
   b. [X] has been communicated by the International Bureau.
   c. [ ] is not required, as the application was filed in the United States Receiving Office (RO/US).
6. [ ] An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
7. [X] Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))
   a. [ ] are transmitted herewith (required only if not transmitted by the International Bureau).
   b. [ ] have been communicated by the International Bureau.
   c. [ ] have not been made; however, the time limit for making such amendments has NOT expired.
   d. [X] have not been made and will not be made.
8. [ ] An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
9. [ ] An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).
10. [ ] An English language translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).

Items 11. to 16. below concern document(s) or information included:
11. [ ] An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
12. [ ] An Assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. [X] A FIRST preliminary amendment
    [ ] A SECOND or SUBSEQUENT preliminary amendment.
14. [ ] A substitute specification.
15. [ ] A change of power of attorney and/or address letter.
16. [X] Other items or information:
    [X] Courtesy copy of the International Application as filed.
    [X] Courtesy copy of the first page of the International Publication (WO 01/38969).
    [X] Formal drawings, 2 sheets, Figures 1-2.
    [X] Application Data Sheet.
    [X] The application is (or will be) assigned to: ENAVIS NETWORKS LTD., whose address is 16 Martin Gehl Street, Petach Tikva 49512, Israel.

U.S. APPLICATION NO (If known, see 37 CFR 1 5) **09/889627**	International Application No PCT/IL00/00769	Attorney's Docket No. ROZENBERG=1

17. [xx] The following fees are submitted:

BASIC NATIONAL FEE (37 CFR 1.492 (a)(1) –(5):

Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO.................$1000.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO...........$860.00

International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO...................$710.00

International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4)...............$690.00

International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4).......................$100.00

CALCULATIONS PTO USE ONLY

				CALCULATIONS (charged)
ENTER APPROPRIATE BASIC FEE AMOUNT =				$ 860.00	
Surcharge of **$130.00** for furnishing the oath or declaration later than [X] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(e)).				$	

Claims as Originally Presented	Number Filed	Number Extra	Rate		
Total Claims	21 - 20	01	X $18.00	$	
Independent Claims	01 - 3		X $80.00	$	
Multiple Dependent Claims (if applicable)			+$270.00	$ 270.00	
TOTAL OF ABOVE CALCULATIONS =				$1,130.00	

Claims After Post Filing Prel. Amend	Number Filed	Number Extra	Rate		
Total Claims	- 20		X $18.00	$	
Independent Claims	- 3		X $78.00	$	
TOTAL OF ABOVE CALCULATIONS =				$1,130.00	

Reduction of ½ for filing by small entity, if applicable. Applicant claims small entity status. See 37 CFR 1 27.	$	
SUBTOTAL =	$1,130.00	
Processing fee of **$130.00** for furnishing the English translation later than [ ] 20 [ ] 30 months from the earliest claimed priority date (37 CFR 1.492(f)).	$	
TOTAL NATIONAL FEE =	$1,130.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00** per property +	$	
TOTAL FEES ENCLOSED =	$1,130.00	

Amount to be: refunded	$
charged	$

a. [ ] A check in the amount of $_____ to cover the above fees is enclosed.
b. [X] Credit Card Payment Form (PTO-2038), authorizing payment in the amount of $ 1,130.00, is attached.
c. [ ] Please charge my Deposit Account No. 02-4035 in the amount of $_____ to cover the above fees. A duplicate copy of this sheet is enclosed.
d. [XX] The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 02-4035. A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

BROWDY AND NEIMARK, P.L.L.C.
624 NINTH STREET, N.W., SUITE 300
WASHINGTON, D.C. 20001
TEL: (202) 628-5197
FAX: (202) 737-3528
Date of this submission. **July 19, 2001**
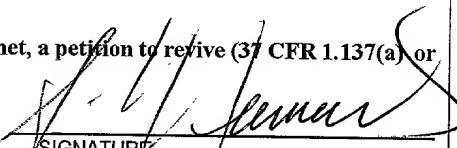
_____ SIGNATURE
Sheridan Neimark
NAME
20,520
REGISTRATION NUMBER

Form PTO-1390 (as slightly revised by Browdy and Neimark)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	Art Unit:
Dominique ROZENBERG et al.)	
)	
IA No.: PCT/IL00/00769)	
)	Washington, D.C.
IA Filed: 19 November 2000)	
)	
U.S. App. No.:)	
(Not Yet Assigned))	
)	July 19, 2001
National Filing Date:)	
(Not Yet Received))	
)	
For: METHOD FOR AUTOMATICALLY…)	Docket No.:
	ROZENBERG=1

PRELIMINARY AMENDMENT

Honorable Commissioner for Patents and Trademarks
Washington, D.C. 20231

Sir:

Contemporaneous with the filing of this case and prior to calculation of the filing fee, kindly amend as follows:

IN THE SPECIFICATION

After the title please insert the following paragraph:

REFERENCE TO RELATED APPLICATIONS

--The present application is the national stage under 35 U.S.C. 371 of international application PCT/IL00/00769, filed which designated the United States, and which international application was published under PCT Article 21(2) in the English language.--

REMARKS

The above amendment to the specification is being made to insert reference to the PCT application of which the present case is a U.S. national stage

Favorable consideration and allowance are earnestly solicited.

Respectfully submitted,
BROWDY AND NEIMARK, P.L.L.C.
Attorneys for Applicant
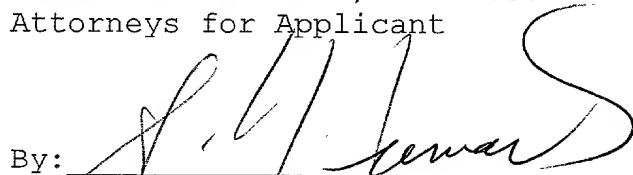
By: _____
Sheridan Neimark
Registration No. 20,520

SN:wrd
Telephone No.: (202) 628-5197
Facsimile No.: (202) 737-3528

METHOD FOR AUTOMATICALLY CONSTRUCTING A COMMAND INTERFACE

FIELD AND BACKGROUND OF THE INVENTION

The present invention relates to a method for constructing an interface for a network device and/or set of network elements for operation by a human user, and in particular, for such a method in which the interface is at least semi-automatically constructed by a software program, thereby significantly reducing the amount of intervention required by a human programmer.

Local area networks (LAN) are widely used for communication between computers which are relatively physically close to each other, or "local". Depending upon the type of network protocol according to which network traffic is directed, various types of hardware devices are used to physically connect each computer to a transport medium, such as a wire. The computers can then communicate by exchanging data. For example, in a client-server system, a plurality of clients could request services from a server. In order for the devices on a LAN to communicate with computers or other devices which are physically farther away, the LAN must be connected to a WAN (wide area network), the Internet or some other type of network.

2

All of these connections require specialized hardware devices, known as network elements. These network elements must be recognized and managed by network management systems. The interaction of the user with such complex systems requires the user to select and execute commands in a

5   logical manner, such that values for parameters for each command are permissible and such that the commands themselves are executed in an order which is both logical and suitable for the complex system. Therefore, the user must be presented with a GUI (graphical user interface) which only enables permitted commands to be selected, and which prevents the user

10   from entering forbidden and/or illogical values for the parameters for each command.

According to the background art, each such interface must be designed and implemented manually, by a human programmer. However, this solution is very time consuming and tedious, as it requires substantially

15   all of the work of implementation to be performed by the human programmer. Furthermore, adding new commands to such a manually constructed interface is a difficult and time consuming process, as these new commands must also be added manually.

Figure 1 shows an exemplary GUI for operating a particularly

20   complex network element system, known as a T::DAX® digital cross connect. The T::DAX® digital cross-connect device is like a switch. According to one functionality thereof, an incoming signal is received from a so-called "from port", and an outgoing signal is transmitted through a

so-called "to port". The cross-connect must be defined in order to pass the

signal from the "from port" to the "to port". The number of elements in a

typical T::DAX® digital cross-connect is very large, even up to thousands of

network elements, such that manual operation of the associated commands is

5   extremely difficult and in some cases almost impossible. Therefore, a GUI

(graphical user interface) is required in order to assist the user to manage the

operation of the T::DAX® digital cross-connect.

According to the background art, a GUI 10 enables the user to select a

type of network element 12, a location of the element 14 and a type of

10   operation 16. Once these choices have been made, then the user may select

one of a plurality of commands from a command menu 18. After selecting

the command, the user then must enter values for various parameters in a

parameter entry sub-window 20, shown here with a mixture of check boxes,

pull-down menus and manual entry boxes, it being understood that the

15   mixture of elements for entering parameter values would clearly depend

upon the parameters themselves.

The significant drawback of background art methods for constructing

GUI 10 is that parameter entry sub-window 20 has to be built separately and

manually for each command, clearly a repetitive and tedious task.

20   Furthermore, each type of operation 16 must be programmed separately.

Since these tasks are repetitive, a significant amount of time is wasted for

constructing GUI 10.

A more useful solution would be generic to commands for any

complex system, and would be built in a modular fashion, such that new commands could be easily accommodated. At the minimum, such a solution would enable at least a portion of the interface to be constructed automatically, thereby reducing the amount of work required for the human

5   programmer. Furthermore, the solution would also organize the commands, thereby producing a GUI display which is simpler to use, as only relevant and suitable commands would be displayed to the user in each screen. Unfortunately, such a solution is not currently available.

Therefore, there is an unmet need for, and it would be highly useful to

10   have, a method for automatically constructing a GUI for interacting with a complex system, which is able to parse information concerning each individual command and to determine the logical relationships between commands, and which is therefore more efficient.

15   SUMMARY OF THE INVENTION

The method of the present invention enables an interface for operating a complex network system to be automatically constructed, by constructing a screen for each command. The commands are read from a command file. Additional files specify the relationships between commands, including

20   commands which can be performed for each operation; values which are excluded and which therefore cannot be entered by the user; related help files for providing assistance to the user concerning these commands; parameter names which are network elements; and at least one command file

concerning specific information for each command. These files are fed into

a parser, which generates at least one metafile for specifying the structure of

the GUI (graphical user interface) to be generated for each command. The

interfaces are then linked to form the overall interface for the plurality of

5   commands. Thus, the method of the present invention enables an interface

for a complex system to be generated from a plurality of modular command

interfaces.   _

According to the present invention, there is provided a method for

automatically generating an interface software program for entering a

10   plurality of commands for operating a complex system by a user, the

complex system featuring a plurality of components each supporting a

sub-plurality of the commands, the steps of the method comprising:

(a) providing a component interface for each of the plurality of

components, the component interface including a command description of

15   each of the corresponding sub-plurality of commands, including a description

of each parameter for each command;

(b) parsing the command description from the component interface to

form a description of a command interface for each command for display to

the user;

20   (c) building the command interface for each command; and

(d) creating the interface software program for displaying the

command interface for each command to the user.

It should be noted that step (a) may be considered as a step of

6

obtaining a ready "modular" component interface being suitable for

combining with other "modular" component interfaces (for example, if all

the components are network elements with TL1 interfaces). In such a module

interface, relationship between the commands and the components, as well as

5   between the commands and operations is known in advance.

However, if at least one component of the complex system does not

have a suitable component interface, step (a) should comprise the following

sub-steps for creating module interfaces for such components:

(i) providing a command description of each command supported by

10   any such component;

(ii) providing a relationship between each said command and each of

such components.

Hereinafter, the term "file" describes a particular set of data which is

not necessarily present as a physically separate file or other data storage unit,

15   but which is described as a file for the sake of convenience.

Hereinafter, the term "computing platform" refers to a particular

computer hardware system or to a particular software operating system.

Examples of such hardware systems include those with any type of suitable

data processor, such as a computer. Hereinafter, the term "computer"

20   includes, but is not limited to, personal computers (PC) having an operating

system such as DOS, Windows™, OS/2™ or Linux; Macintosh™ computers;

computers having JAVA™-OS as the operating system; and graphical

workstations such as the computers of Sun Microsystems™ and Silicon

Graphics™, and other computers having some version of the UNIX operating system such as AIX™ or SOLARIS™ of Sun Microsystems™; a PalmPilot™, a PilotPC™, or any other handheld device; or any other known and available operating system. Hereinafter, the term "Windows™" includes but is not

5 limited to Windows95™, Windows 3.x™ in which "x" is an integer such as "1", Windows NT™, Windows98™, Windows CE™ and any upgraded versions of these operating systems by Microsoft Corp. (USA).

For the present invention, a software application could be written in substantially any suitable programming language, which could easily be

10 selected by one of ordinary skill in the art. The programming language chosen should be compatible with the computer according to which the software application is executed. Examples of suitable programming languages include, but are not limited to, C, C++ and Java. Furthermore, the functions of the present invention, when described as a series of steps for a

15 method, could be implemented as a series of software instructions for being operated by a data processor, such that the present invention could be implemented as software, firmware or hardware.

BRIEF DESCRIPTION OF THE DRAWINGS

20 The invention is herein described, by way of example only, with reference to the accompanying drawings, wherein:

FIG. 1 is an illustrative background art GUI (graphical user interface)

for executing a plurality of commands for controlling a complex system; and

FIG. 2 is a flowchart of an exemplary method for constructing an

interface according to the present invention.

5   DESCRIPTION OF THE PREFERRED EMBODIMENTS

The method of the present invention enables a command interface for

a network element to be automatically constructed from a modular command

description, such that interlinking command interfaces can be created from a

plurality of command descriptions. Since each command description is

10   modular, interfaces for new commands can be easily added and integrated

into an existing system of interfaces for interacting with the network element.

The method of the present invention enables a set of interfaces to be more

easily constructed and updated for a complex system such as a T::DAX®

digital cross-connect. Furthermore, these interfaces present commands

15   grouped in a logical manner, such that each interface only shows the user

commands which are available at each stage of the interaction, thereby

simplifying the complexity of the management of these commands. Thus,

the method of the present invention enables a set of interfaces to be

automatically created and updated, which both simplifies the task of

20   programming these interfaces and of operating the interfaces once they have

been created.

According to one embodiment of the present invention, if each

component interface for a particular network element, or component, is able

to interact with other such component interfaces, then the GUI system may

be built from such component interfaces. Each component interface includes

a command description of each of the plurality of commands. One example

of such component interfaces which are suitable as modular building blocks

5   is the TL1 type of interface.

Alternatively, if the component interfaces are either not available or

else are not suitable as modular building blocks, then the GUI system is built

by determining both the command description for each command and the

relationship between commands, without reliance upon the component

10   interfaces.

There is, of course, yet another and most practical option which

constitutes a combination of the two above versions and comprises

partially building the GUI system with respect to a part of the components

from their component interfaces if readily available and suitable, and

15   thereupon

completing the GUI system by determining the command description for

each command and the relationship between commands and components

with respect to the remaining components.

The principles and operation of the present invention may be better

20   understood with reference to the drawings and the accompanying description.

Referring now to the drawings, Figure 2 is a flowchart of an

exemplary method for creating a set of interfaces for executing a plurality of

commands. This set of interfaces is also referred to herein as the "resultant

10

GUI". The method is explained with regard to a T::DAX® digital cross connect, the resultant background art GUI for which is shown and described above with regard to Figure 1, it being understood that this is for the purposes of description only and is not meant to be limiting in any way, as

5   the method of the present invention would be suitable for operation with any complex system containing a plurality of commands.

As shown, in step 1, a command file is created. The command file contains information concerning at least one command, although preferably information concerning a plurality of commands is contained in the

10   command file. The command file contains a description of the parameters for each command, default values for each parameter, and a range of permissible values for each parameter. If the command file includes information for only one command, then a plurality of command files must be created, one for each command or group of commands. The description

15   of each parameter should also include the type of values expected for entry by the user, such as binary (off/on), a character string, a number or other description of the command parameter values.

In step 2, at least one overall system file is created, which defines relationships between different features of the system. Preferably, a plurality

20   of separate system files are created, although as previously noted, each separate "file" describes a particular set of data which is not necessarily stored in a physically separate file or other physical data storage unit. Rather, a plurality of sets of data could be concatenated into a single physical

data storage unit, such as a single file for example. Each overall system file

may be created manually, preferably with the assistance of a software

program for file creation.

One such overall system file is a list of operations and commands

5   which are permitted for each component of the complex system. For

example, for a T::DAX® cross connect, each component would be a network

element. In addition, this component file preferably also includes a

description of the relationships between the commands and the operations,

such that for each operation, a set of at least one permitted command is

10   described. Preferably, the component file also includes a "security gate", or

description of the security clearance which a user must have in order to

execute each command for each component.

The next preferred overall system file is an excluded values file,

which lists values which are neither shown as choices to the user through the

15   resultant GUI, nor are these values accepted from the user from the resultant

GUI. These excluded values are typically commands or values for

debugging the resultant GUI and/or the operation of a component within the

complex system. More preferably, the list of excluded values optionally

decreases the number of permitted values from those given in the command

20   file.

Another preferred overall system file is a list of help files, for

providing assistance to the user for each command. Preferably, a help file is

listed for each command and for each element type. More preferably, the

12

help file is made accessible to the user through the resultant GUI once a

command is highlighted or otherwise indicated, by clicking or otherwise

selecting an appropriate GUI element. Most preferably, a separate help file

or at least separate help file information is provided for each combination of

5   a network element type and a command.

Yet another preferred overall system file is a network elements file,

which contains a list of parameter values which are actually network

elements. The network elements file is created manually, as for the other

overall system files.

10   In step 3, the overall system files and the command file are parsed in

order to generate at least one, and preferably a plurality of, separate

metafiles. The first such metafile is a GUI builder metafile. A GUI builder

is a software program which creates a GUI from input information and which

is known in the art. The GUI builder metafile is preferably suitable for

15   interaction with a commercially available GUI builder software program,

such as ILOG View™ (ILOG SA, France). The GUI builder metafile

preferably features information for one command, in which groups of

acceptable parameters and other information is given. More preferably, these

groups of acceptable parameters are organized into blocks, with each block

20   being identified as either a parameter or a network element type. The

identification of each block is most preferably performed according to the

information contained in the network elements file, as previously described.

The GUI builder file is an input file to a specialized program for building

ILOG screens, such as a commercially available GUI builder software

program, as previously described.

The second such metafile is a GUI application file for generating the

underlying application for actually operating the GUI. The GUI application

5 file contains the supported types of network elements for each command,

preferably separated by category of command.

In addition, the GUI application file preferably has a list of parameters

which may be manipulated by the user through the resultant GUI when the

underlying application is being operated by the computer, or "running". The

10 GUI application file is very similar to the previously described GUI builder

metafile, except that the GUI application file includes the parameters and

values for each command. These parameters are the fields displayed in the

resultant GUI.

Preferably, each parameter is described with the name and type of

15 parameter, optionally the block to which the parameter belongs if this is

applicable, a default value for the parameter, and a description of the

permissible values for the parameter. The description of the permissible

values is optionally given either as minimum and maximum values for a

numerically valued parameter, or alternatively as a list of names of

20 acceptable values for a character string.

Preferably, the step of parsing the GUI application file includes the

step of converting the organized text layout to header files for an underlying

generic software program which actually runs the GUI system. These header

files are created by the software program incorporating the method of the present invention, and are built from the GUI application metafile. The header files are then read by the underlying software program for operating the resultant GUI. This step is preferred since it enables generic software to

5   be reused for many different GUI systems.

Each header file preferably includes a list of the different types of network elements, or other components of the complex system such as the T::DAX® digital cross connect; a relationship between each type of element and a category of commands. a list of commands with a description of each

10   command; a list of parameters for each command; and a security gate, which is security information for each command.

In step 4, a predefined template for creating the GUI is supplied. This predefined template preferably features the names of the fields of parameter values which are entered by the user, such as "from:" and "to:", as well as

15   the basic layout of the GUI elements. In addition, the predefined template also includes words for describing each command, such as "RTRV-CRS". The words which are displayed are more preferably stored in a Data Dictionary, thereby enabling the user to change the words to be displayed. For example, the words for the command "RTRV-CRS" could be changed to

20   display "Retrieve CRS". Furthermore, at run time preferably the list of network elements which are addressable is optionally and preferably displayed, thereby enabling the user to separately perform a command on a single network element in the system.

The actual, full T::DAX command name is RTRV-CRS-element type,

such that if the element type is E1, then the command is given as

"RTRV-CRS-E1", and so forth. The element type is displayed in the

resultant GUI, with the command name. However, for clarity, optionally the

5   "elementType" field is not displayed with the command name itself, but

rather in a separate display.

In step 5, a GUI software program which is specific for operating the

GUI system is created. The GUI software program receives the header file

for each command, created as described above, in order to operate the GUI

10   for each command.

Thus, the method of the present invention overcomes the limitations

of the background art methods, by enabling a set of interfaces to be

automatically created as a GUI system for operating a complex system such

as a T::DAX® digital cross connect, as well as for all devices which use the

15   TL1 interface. In order to be adapted to these other devices, the present

invention need only receive a command file which describes each command

and the associated parameter(s), or alternatively the previously described

metafiles, as input. The method of the present invention is particularly

suitable for operation with the TL1 interface, as each command of this

20   interface features several parameters and values. Therefore, a GUI system is

generally suitable and user-friendly for user interactions with the TL1

commands. The method of the present invention facilitates such a GUI

system by enabling the system to be automatically created, which is more

efficient than manual creation of such a set of interfaces.

Indeed, on a more general level, the method of the present invention only requires a description of each command, including a description of the parameter(s) for each command, with a description of the relationships

5 between commands and the components of the complex system, in order to build the GUI system as an interface for the user. Thus, the present invention is highly adaptable to various types of GUI systems for operating such complex systems.

While the invention has been described with respect to a limited

10 number of embodiments, it will be appreciated that many variations, modifications and other applications of the invention may be made.

CLAIMS:

1.   A method for automatically generating an interface software program for entering a plurality of commands for operating a complex system by a user, the complex system featuring a plurality of components

5   each supporting a sub-plurality of the commands, the steps of the method comprising:

(a)   providing a component interface for each of the plurality of components, said component interface including a command description of each of the corresponding sub-plurality of

10   commands, including a description of each parameter for each command;

(b)   parsing said command description from said component interface to form a description of a command interface for each command for display to the user;

15   (c)   building said command interface for each command; and

(d)   creating the interface software program for displaying said command interface for each command to the user.

2.   The method of claim 1, wherein for at least a sub-plurality of the components, step (a) comprises the steps of:

20   (i)   providing a command description of each command supported by any of said sub-plurality of the components;

(ii)   providing a relationship between each said command and each of said sub-plurality of the components.

3.   The method of claim 1 or 2, wherein the interface software

program is generated for a complex system for interacting with a plurality of

network elements, each network element corresponding to a component of

the complex system.

5

4.   The method of claim 3, wherein the complex system is a

T::DAX® digital cross connect .

5.   The method according to any one of claims 1 to 4, wherein step

10   (a) further includes the steps of:

(iii)   providing a default value for each parameter; and

(iv)   providing a plurality of permissible values for each parameter.

6.   The method of claim 5, wherein said plurality of permissible

15   values includes a minimum value and a maximum value for numeric

parameters, and a list of names for character string parameters.

7.   The method of claim 5, wherein sub-steps (iii) and (iv) of step

(a) further comprise the steps of determining said default value and said

20   plurality of permissible values for each parameter for each of the plurality of

components.

8. The method of claim 7, wherein sub-step (iv) of step (a) further comprises the step of providing a list of excluded values for each parameter of each command.

5　　9. The method of claim 8, wherein the interface software program is generated for a complex system for interacting with a plurality of network elements, each network element corresponding to a component of the complex system and wherein step (a) further comprises the step of providing a list of at least one parameter corresponding to a network element.

10

10. The method of claim 2, wherein sub-step (ii) of step (a) further comprises the steps of:

(A) determining at least one operation for performing with each of the plurality of components; and

15　　(B) determining a relationship between said at least one operation and each of the plurality of commands.

11. The method of claim 10, wherein step (a) further comprises the step of:

20　　(C) determining a security clearance of the user required before a command is accessed by the user.

20

12.   The method of claim 1, wherein step (a) further comprises the

step of determining a help file for each of the plurality of commands, said

help file including information for assisting the user for each command.

5   13.   The method of claim 12, wherein the interface software

program is generated for a complex system for interacting with a plurality of

network elements, each network element corresponding to a component of

the complex system and wherein said help file is provided for each

combination of a network element and a command for operation with said

10   network element.

14.   The method of claim 1, wherein step (b) further comprises the

step of determining a template for said interface.

15   15.   The method of claim 14, wherein said interface is a GUI

(graphical user interface) and said template features a plurality of GUI

elements.

16.   The method of claim 15, wherein said template features a name

20   for each field corresponding to each parameter.

21

17.   The method of claim 16, wherein said template features a name

for each command, said name being altered according to a selection by the

user.

5   18.   The method of claim 15, wherein step (c) is performed by a

GUI builder software program.

19.   The method of claim 1, wherein step (d) further comprises the

steps of:

10   (i)   providing a generic interface operation software program;

(ii)   generating a header file for each command; and

(iii)   constructing the interface software program from said header

file and said generic interface operation software program.

15

ABSTRACT

A method for enabling a command interface for a
network element to be automatically constructed from a modular
command description, such that interlinking command interfaces
can be created from plurality of command descriptions. Since
each command description is modular, interfaces for new
commands can be easily added and integrated into an existing
system of interfaces for interacting with the network element.
The method enables a set of interfaces to be more easily
constructed and updated for a complex system such as a T::DAX®
digital cross-connect. Furthermore, these interfaces present
commands grouped in a logical manner, such that each interface
only shows the user commands which are available at each stage
of the interaction, thereby simplifying the complexity of the
management of these commands. Thus, the method enables a set
of interfaces to be automatically created and updated, which
both simplifies the task of programming these interfaces and
of operating the interfaces once they have been created.

1/2

FIG.1

CREATE A COMMAND FILE ⟍ 1

CREATE AT LEAST ONE OVERALL SYSTEM FILE ⟍ 2

PARSE THE AT LEAST ONE OVERALL SYSTEM FILE AND THE COMMAND FILE TO GENERATE AT LEAST ONE METAFILE ⟍ 3

SUPPLY A PREDEFINED TEMPLATE FOR THE INTERFACE ⟍ 4

CREATE AN INTERFACE SOFTWARE PROGRAM FOR DISPLAYING THE INTERFACE ⟍ 5
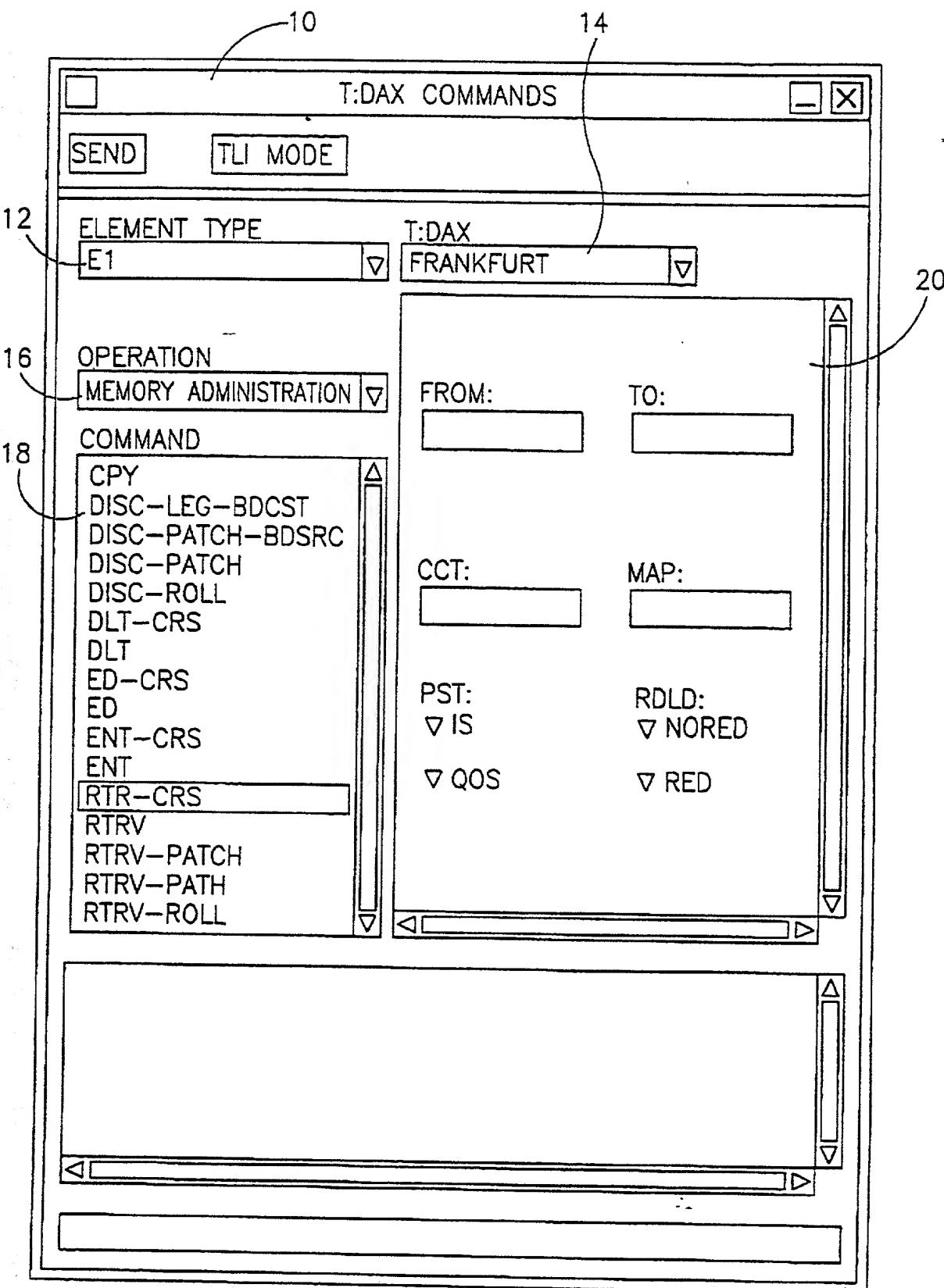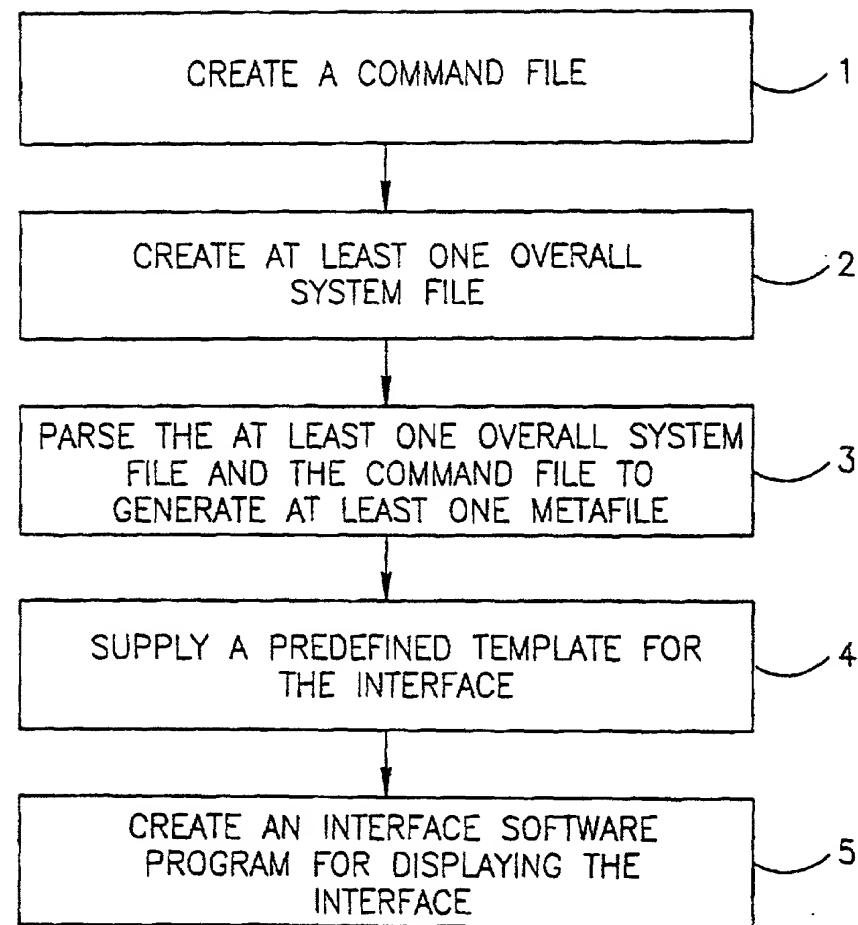
FIG.2

Page 1 of 2 Pages   [ ] Original   [ ] Supplemental   Atty. Docket:

Combined Declaration for Patent Application and Power of Attorney

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; and that I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

Method for automatically constructing a command interface

[ ]   is attached hereto;

[ ]   was filed in the United States under 35 U.S.C. '111 on May 1, 2001 as U.S. Appln. No. _____*; or

[X]   was/will be filed in the U.S. under 35 U.S.C. '371 by entry into the U.S. national stage of an international (PCT) application, PCT/IL00/00769; filed on November 19, 2000, entry requested on _____*; national stage application received U.S. Appln. No. _____*; '371/'102(e) date _____*

(* if known)

and was amended on _____ (if applicable).

(include dates of amendments under PCT Art. 19 and 34 if PCT)

I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above; and I acknowledge the duty to disclose to the Patent and Trademark Office (PTO) all information known by me to be material to patentability as defined in 37 C.F.R. '1.56.

I hereby claim foreign priority benefits under 35 U.S.C. " 119 and 365 of any prior foreign application(s) for patent or inventor=s certificate, or prior PCT application(s) designating a country other than the U.S., listed below with the AYes@ box checked and have also identified below any such application having a filing date before that of the application on which priority is claimed

133082	Israel	22/11/1999	[X]	[]
(Number)	(Country)	(Day Month Year Filed)	YES	NO
			[]	[]
(Number)	(Country)	(Day Month Year Filed)	YES	NO

I hereby claim the benefit under 35 U.S.C. '120 of any prior U.S. non-provisional application(s) or prior PCT application(s) designating the U.S. listed below, or under '119(e) of any prior U.S. provisional applications listed below, and, insofar as the subject matter of each of the claims of this application is not disclosed in such U.S. or PCT application in the manner provided by the first paragraph of 35 U.S.C. '112, I acknowledge the duty to disclose to the PTO all information as defined in 37 C.F.R. '1.56(a) which occurred between the filing date of the prior application and the national filing date of this application:

(Application No.)	(Day Month Year Filed)	(Status: patented, pending, abandoned)
(Application No.)	(Day Month Year Filed)	(Status: patented, pending, abandoned)
(Application No.)	(Day Month Year Filed)	(Status: patented, pending, abandoned)

As a named inventor, I hereby appoint the following registered practitioners to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

All of the practitioners associated with Customer Number 001444

Direct all correspondence to the address associated with Customer Number 001444, i.e.,

BROWDY AND NEIMARK, P.L.L.C.
624 Ninth Street, N.W.
Washington, D.C. 20001-5303
(202) 628-5197

The undersigned hereby authorizes the U.S. Attorneys or Agents appointed herein to accept and follow instructions from _____ as to any action to be taken in the U.S. Patent and Trademark Office regarding this application without direct communication between the U.S. Attorneys or Agents and the undersigned. In the event of a change of the persons from whom instructions may be taken, the U.S. Attorneys or Agents appointed herein will be so notified by the undersigned.

Page 2 of 2 Pages   Atty. Docket:

Title: Method for automatically constructing a command interface

U.S. Application filed on _____ , Serial No._____

PCT Application filed on November 19, 2000, Serial No. PCT/IL00/00769

I hereby further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. '1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

FULL NAME OF FIRST INVENTOR Dominique ROZENBERG	INVENTOR=S SIGNATURE	DATE 11/06/01
RESIDENT 7/11 Ahim Kibovitch Street Rehovot 76450 – Israel ILX		CITIZENSHIP Israeli
POST OFFICE ADDRESS as above		
FULL NAME OF SECOND JOINT INVENTOR Felix FRUHTMAN	INVENTOR=S SIGNATURE	DATE 11/6/01
RESIDENT 9/11 Beit Lehem Street Rishon Lezion 75323 - Israel ILX		CITIZENSHIP Israeli
POST OFFICE ADDRESS as above		
FULL NAME OF THIRD JOINT INVENTOR Eytan SHAHAM	INVENTOR=S SIGNATURE	DATE 10/9/01
RESIDENT 1 Sorek Street Rosh Ha'Ayin 48571 – Israel ILX		CITIZENSHIP Israeli
POST OFFICE ADDRESS as above		
FULL NAME OF FOURTH JOINT INVENTOR	INVENTOR=S SIGNATURE	DATE
RESIDENT		CITIZENSHIP
POST OFFICE ADDRESS		
FULL NAME OF FIFTH JOINT INVENTOR	INVENTOR=S SIGNATURE	DATE
RESIDENT		CITIZENSHIP
POST OFFICE ADDRESS		
FULL NAME OF SIXTH JOINT INVENTOR	INVENTOR=S SIGNATURE	DATE
RESIDENT		CITIZENSHIP
POST OFFICE ADDRESS		

ALL INVENTORS MUST REVIEW APPLICATION AND DECLARATION BEFORE SIGNING. ALL ALTERATIONS MUST BE INITIALED AND DATED BY ALL INVENTORS PRIOR TO EXECUTION. NO ALTERATIONS CAN BE MADE AFTER THE DECLARATION IS SIGNED. ALL PAGES OF DECLARATION MUST BE SEEN BY ALL INVENTORS.